Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) A method for processing media data, the method comprising:

transferring, by a stream-source object, multimedia data to a control layer, which provides a single interface between the stream-source object and a plurality of other objects, wherein the stream-source object operates subject to control of a media-source object;

receiving at the control layer the multimedia data, which includes a plurality of media data streams ~~at once~~ in different formats ~~via a control layer~~;

transferring a media data stream of the plurality of media data streams to a transform object, which converts the media data stream into a format that is usable by a data sink;

receiving in the control layer the media data stream, which has been converted into the format; and

transferring the media data stream to a stream-sink object, which is usable to transfer the media data stream to the data sink,

(1) wherein the stream-sink object operates subject to a media-sink object that implements a state machine ~~modifying the media data streams in one or more stream sinks; implementing in a media sink one or more state machines~~ to control a state of transfer of the media data stream[[s]] on a per stream basis, the ~~one or more~~ state machine[[s]] being

implemented according to a control signal ~~one or more control signals~~ from the control layer, and ~~the media sink providing a common interface for processing the media data streams in different formats; and using the state of the media data streams to modify the functionality of the stream sinks,~~

(2) wherein the control layer signals the stream-sink object that a discontinuity exists in another media data stream ~~one or more stream sinks that one or more discontinuities exist in one or more media data streams~~ by placing an associated marker in the other ~~one or more~~ media data stream[[s]].

2. (Currently Amended) The method of claim 1 wherein a [[the]] modification of the media data stream[[s]] in the stream-sink object is dynamic.

3. (Currently Amended) The method of claim 1 further comprising:

throttling processing of the media data stream[[s]] via the stream-sink object ~~stream sinks~~ based on the media-sink object ~~one or more media sink components~~.

4. (Currently Amended) The method of claim 1 wherein one or more of the media-sink object and the stream-sink[[s]] object provide notifications of [[for]] events to the control layer.

5. (Currently Amended) The method of claim 1 wherein the media data stream switches to a second media-sink object upon a detection of invalid media sink.

6. (Currently Amended) The method of claim 1 wherein the media-sink object directs multiplexing of two or more of the media data streams into a same media-sink object.

7. (Currently Amended) The method of claim 1 wherein the control layer directs control and timing of [[for]] the media-sink object and the stream-sink[[s]] object.

8. (Currently Amended) The method of claim 1 wherein the control layer directs format negotiation to be performed in the stream-sink[[s]] object, the format appropriate for an output device.

9. (Currently Amended) The method of claim 1 wherein the control layer includes a media engine and a media processor, the media engine communicating with a core layer to direct a pipeline through the transform object and to the media-sink object ~~one or more multimedia transforms and to the media sink~~.

10. (Currently Amended) The method of claim 9 wherein the core layer includes the media-sink object, the stream-sink object ~~one or more stream sinks~~, the media-source object ~~a media source~~, the transform object, and the stream-source object ~~multimedia transforms and stream sources~~.

11. (Currently Amended) The method of claim 1 wherein the stream-sink object accesses an application programming interfaces (API) that enables the stream-sink object to access a pointer to the media-sink object.

12. (Currently Amended) The method of claim 1 wherein the stream-sink object accesses an application programming interfaces (API) that provides an identifier for the media-sink object.

13. (Currently Amended) The method of claim 1 wherein the stream-sink object accesses an application programming interfaces (API) that provides a type of media in use.

14. (Currently Amended) The method of claim 1 wherein the stream-sink object accesses an application programming interfaces (API) configured to cause processing of a sample of the media data.

15. (Currently Amended) The method of claim 1 wherein the stream-sink object accesses an application programming interfaces (API) configured to remove any data that has not been processed.

16. (Currently Amended) The method of claim 1 wherein the stream-sink object accesses an application programming interfaces (API) configured to place a marker in the data stream to determine when the stream-sink object has finished processing received data associated with the marker.

17. (Currently Amended) The method of claim 1 wherein the stream-sink object accesses an application programming interfaces (API) configured to identify an end of a segment of the media data.

18. (Previously Presented) The method of claim 9 wherein the core layer is configured to communicate to retrieve characteristics of a sample allocator.

19. (Previously Presented) The method of claim 9 wherein the core layer is configured to request that a sample allocator acquire any needed resources.

20. (Previously Presented) The method of claim 9 wherein the core layer is configured to request that a sample allocator end an asynchronous resource allocation process.

21. (Previously Presented) The method of claim 9 wherein the core layer is configured to request that a sample allocator retrieve one or more of a maximum number of samples in a sample allocation and any requested samples.

22. (Previously Presented) The method of claim 9 wherein the core layer is configured to request that a sample allocator cancel one or more allocations.

23. (Currently Amended) A computer storage ~~readable~~ medium having computer-executable instructions stored thereon that, when executed, cause a computing device to perform a method for processing data through a collection of one or more media objects, the computer-executable instructions performing acts comprising:

receiving a plurality of media data streams at once in different formats via a control layer;

transferring a media data stream of the plurality of media data streams to a transform object to be modified; ~~modifying the media data streams in one or more stream sinks;~~

receiving at the control layer the one of the media data streams that has been modified;

implementing in a media sink a state machine ~~one or more state machines~~ to control a state of transfer of the media data streams on a per stream basis, the ~~one or more~~ state machine[[s]] being implemented according to a ~~one or more~~ control signal[[s]] from the control layer, ~~the media sink providing a common interface for processing the media data streams in different formats~~; and

throttling the progress of the plurality of media data streams at a stream sink by controlling a rate of allocation.

~~using the state of the media data streams to modify the functionality of the stream sinks,~~

~~wherein the control layer signals the one or more stream sinks that one or more discontinuities exist in one or more media data streams by placing an associated marker in the one or more media data streams.~~

24. (Currently Amended) The computer readable medium of claim 23 wherein [[the]] modification of the data streams in the stream sink is dynamic.

25. (Canceled)

26. (Currently Amended) The computer readable medium of claim 23 wherein one or more of the media sink and the stream sink[[s]] provides notifications of [[for]] events to the control layer.

27. (Original) The computer readable medium of claim 23 wherein the media data stream switches to a second media sink upon a detection of invalid media sink.

28. (Original) The computer readable medium of claim 23 wherein the media sink directs multiplexing of two or more of the media data streams into a same media sink.

29. (Currently Amended) The computer readable medium of claim 23 wherein the control layer directs control and timing of [[for]] the media sink and the stream sinks.

30. (Original) The computer readable medium of claim 23 wherein the control layer directs format negotiation to be performed in the stream sinks, the format appropriate for an output device.

31. (Original) The computer readable medium of claim 23 wherein the control layer includes a media engine and a media processor, the media engine communicating with a core layer to direct a pipeline through one or more multimedia transforms and to the media sink.

32. (Currently Amended) A computing system, which includes one or more computing devices having a processor, a server, and a computer storage medium, the computing system configured to provide a multimedia system, the multimedia system comprising:

a control layer maintained on one or more computing devices, the control layer configured to provide a single interface to a media source, a transform, a media sink, and a stream sink; and ~~receive a plurality of media data streams at once in different formats from an application;~~

a core layer maintained on the one or more computing devices, the core layer coupled to the control layer, the core layer including:

(1) the media source, which transfers media data streams to the control layer;

(2) the transform, which receives the media data streams from the control layer and converts a format of the media data streams;

(3) the media sink ~~one or more media sink components~~ configured to implement a state machine ~~one or more state machines~~ to control transfer of the media data streams on a per stream basis through the multimedia system, the ~~one or more~~ state machine[[s]] being implemented according to a ~~one or more~~ control signal[[s]] from the control layer~~, the media sink components providing a common interface for processing the media data streams in different formats~~; and

(4) a stream sink ~~one or more stream sinks~~ configured to dynamically modify the media data streams via the control layer and an identified state of the media data streams determined in the media sink components~~,~~

~~wherein the control layer signals the one or more stream sinks that one or more discontinuities exist in one or more media data streams by placing an associated marker in the one or more media data streams~~.

33. (Currently Amended) The computing ~~multimedia~~ system of claim 32 wherein the control layer is an application programming interface (API).

34. (Currently Amended) The computing ~~multimedia~~ system of claim 32 wherein the control layer includes a media engine and a media processor, the media engine communicating with a core layer to direct a pipeline through one or more multimedia transforms and to the media sink.

35. (Currently Amended) The computing ~~multimedia~~ system of claim 32 wherein the core layer includes the media sink, the stream sinks, [[a]] the media source, the transform, and a stream source. ~~one or more multimedia transforms and one or more stream sources.~~

36. (Currently Amended) The computing ~~multimedia~~ system of claim 32 wherein the core layer is configured to communicate with the media sink to retrieve the characteristics of the media sink.

37. (Currently Amended) The computing ~~multimedia~~ system of claim 32 wherein the core layer is configured to communicate with the media sink to add an additional stream sink and remove one of the stream sinks.

38. (Currently Amended) The computing ~~multimedia~~ system of claim 32 wherein the core layer is configured to communicate with the media sink, the media sink enabled to report the number of stream sinks associated with a given media sink.

39. (Currently Amended) The computing ~~multimedia~~ system of claim 32 wherein the core layer is configured to communicate with the stream sinks to send a pointer to a stream sink associated with the media sink by an index in the media sink.

40. (Currently Amended) The computing ~~multimedia~~ system of claim 32 wherein the core layer is configured to communicate to send a pointer, to a stream sink associated with the media sink using a stream sink identifier.

41. (Currently Amended) The computing ~~multimedia~~ system of claim 32 wherein the core layer is configured to communicate to set a rate of a presentation clock and retrieve a presentation clock setting.

42. (Currently Amended) The computing ~~multimedia~~ system of claim 32 wherein the core layer is configured to communicate to retrieve characteristics of a sample allocator.

43. (Currently Amended) The computing ~~multimedia~~ system of claim 32 wherein the core layer is configured to request that a sample allocator acquire any needed resources.

44. (Currently Amended) The computing ~~multimedia~~ system of claim 32 wherein the core layer is configured to request that a sample allocator end an asynchronous resource allocation process.

45. (Currently Amended) The computing ~~multimedia~~ system of claim 32 wherein the core layer is configured to request that a sample allocator retrieve one or more of a maximum number of samples in a sample allocation and any requested samples.

46. (Currently Amended) The computing ~~multimedia~~ system of claim 32 wherein the core layer is configured to request that a sample allocator cancel one or more allocations.